
Software Requirement Specification Pattern Pada Aplikasi Sistem Informasi Registrasi Akademik

Yanti Andriyani

Jurusan Matematika, Fakultas Matematika dan Ilmu Pengetahuan Alam,
Universitas Riau
Kampus Bina Widya Km.125 Simpang Baru, Pekanbaru, Indonesia 28293
e-mail: yeandriyani@yahoo.com

Abstrak

Dunia perguruan tinggi telah memanfaatkan informasi untuk mengefisiensikan proses-proses akademik yang ada di dalamnya yang disebut dengan Sistem Informasi Akademik (SIAK). SIAK dalam kaitannya dengan proses pengembangan perangkat lunak adalah pada tahapan dalam melakukan pembangunan aplikasi SIAK tersebut dimana sebagian besar tahapan pembangunannya dan pengembangannya masih bersifat ad hoc. Jika dicermati, SIAK ini dibangun menurut kebutuhan yang dapat dilihat dari prosesnya. Dari analisis implementasi SIAK, pada beberapa institusi akan diperoleh beberapa proses yang umum. Proses yang umum ini akan merepresentasikan keumuman dalam kebutuhan spesifikasi perangkat lunak. Dalam penelitian ini akan dilakukan analisis pada beberapa aplikasi SIAK yang digunakan pada beberapa institusi pendidikan tinggi untuk dilihat keumuman prosesnya. Fokus penelitian pada registrasi akademik sebagai bagian dari SIAK. Format representasi pattern yang diimplementasikan pada hasil analisis akan mengacu pada penelitian Eko K Budiardjo dengan Software Requirement Specification (SRS) Pattern-nya. SRS pattern yang berhasil diidentifikasi pada penelitian ini adalah Registrasi KRS/IRS Mahasiswa, Perubahan KRS/IRS mahasiswa, dan persetujuan KRS/IRS mahasiswa. Penelitian ini menyimpulkan bahwa SRS pattern bisa direpresentasikan dengan memahami proses registrasi akademik yang sudah diimplementasikan untuk mendapatkan kebutuhan software yang spesifik.

Kata kunci: Software Requirement, Analysis Pattern, Requirement Pattern Software, Requirement Specification Pattern, Course Management, UML

Abstract

Education institution use information system to efficiency academic process, that is called Academic Information System (AIS). AIS related with software development process in stage of build and develop it. A large part build and development stage stills done according to ad hoc. AIS built based on requirement that seen from the process. If we analyze, this process will present set of process which is common in several higher education institutions. So that commonality can be develop as basic requirement and represent as Pattern. This research will analyze several AIS application to find commonality process. The commonality will represented in Pattern form use Software Requirement Specification (SRS) Pattern. It represented analysis result use Software

Requirement Specification (SRS) Pattern refer to Eko K. Budiardjo. SRS Pattern for academic information system application identified in this research is Student KRS/IRS Registration, Editing KRS/IRS, and Approval KRS/IRS. This research also conclude that requirement pattern can be constructed from process or by catching the elementary part from best practice, or by comparing AIS to find commonality process.

Keywords: *Software Requirement, Analysis Pattern, Requirement Pattern Software, Requirement Specification Pattern, Course Management, UML*

1. Pendahuluan

Proses pengembangan perangkat lunak adalah proses yang tidak sederhana dan memerlukan pengetahuan dan keterampilan yang khusus. Dalam bidang ini telah berkembang berbagai teknik dan metodologi yang mempermudah proses pengembangan perangkat lunak yang berkualitas. Proses pengembangan perangkat lunak sendiri terdiri dari beberapa tahapan dengan tahapan umum dimulai dari *analysis-design-coding-testing-implementation*, dengan berbagai variasinya bergantung metodologi pengembangan perangkat lunak tersebut. Dalam proses mengembangkan suatu piranti lunak, pada masalah apapun, akan diawali oleh tahapan analisa kebutuhan yang akan menghasilkan kebutuhan. Hasil analisis untuk pengembangan aplikasi pada suatu domain permasalahan yang sama biasanya akan menghasilkan spesifikasi kebutuhan yang sama. *Requirement* pada suatu domain masalah ini akan memiliki pola (*pattern*) yang sama. *Pattern* adalah sekumpulan pengetahuan eksplisit yang menjelaskan suatu problem dan solusinya dalam suatu konteks tertentu. Dalam bidang *software engineering*, telah banyak dikembangkan *software requirement pattern* (atau sering disebut dengan *analysis pattern*) untuk berbagai domain permasalahan, misalnya pada masalah real-time sistem atau pada aplikasi bisnis. *Pattern* ini biasanya direpresentasikan dalam suatu kerangka penulisan yang disebut dengan *template*.

Dalam penelitian ini akan menggunakan *pattern set* yang telah dikembangkan dalam penelitian Eduardo B. Fernandez yang melakukan penelitian *Pattern* tentang *Analysis Pattern for Course Management* dan direpresentasikan dengan cukup sederhana [1], [2], [3], [4]. Fernandez membagi *Pattern* dalam penelitian ini menjadi dua bagian, yaitu *Course Registration* dan *Grade Management*, yang akhirnya nanti akan digabungkan menjadi *Course Management*. Dua komponen *Pattern* ini memiliki nilai sendiri dan dibahas secara terpisah. Sedangkan format representasi *Pattern* yang digunakan akan mengacu pada hasil penelitian Eko K. Budiardjo dengan *Software Requirement Specification (SRS) Pattern*-nya [5], [6], [7]. Representasi *template SRS Pattern* ini merupakan hasil dari penelitian dari apa yang telah dicapai dari beberapa *Pattern* sebelumnya, *design Pattern*, *analysis Pattern* dan *requirement Pattern*. *Template* ini menggunakan standar notasi UML (*Unified Modeling Language*) yang digunakan untuk merepresentasikan *artifact* dalam tahapan pengembangan dan pembentukan representasi beberapa atribut pada *pattern*.

Penelitian ini akan mencoba untuk melakukan perbandingan beberapa aplikasi Sistem Informasi Akademik (SIK) dari sisi *course registration* (proses registrasi akademik) untuk mendapatkan suatu hal yang bisa dijadikan sebagai *pattern* yang sifatnya umum (*common*) dengan menggunakan representasi *Software Requirement Specification Pattern (SRS Pattern)*. Domain permasalahan ini dipilih

karena permasalahan ini belum terlalu menjadi perhatian dalam pengembangan aplikasi Sistem Informasi Akademik yang sebagian besar masih mengembangkan aplikasi secara *ad hoc*, dan topik *requirement pattern* sebagian besar hanya terfokus pada proses bisnis perusahaan yang sifatnya komersial.

2. Metodologi Penelitian

Penelitian ini fokus pada proses penyusunan representasi *Software Requirement Specification (SRS) Pattern*. Untuk memperoleh hasil akhir yang berupa representasi *Software Requirement Specification (SRS) Pattern*, maka diambil domain permasalahan tentang proses registrasi akademik di sebuah universitas sebagai studi kasus. Sedangkan metodologi yang diimplementasikan pada beberapa studi kasus.

Prosedur penelitian yang dilakukan diawal adalah melakukan studi pustaka/literatur mengenai *software requirement* dan *requirement Pattern* secara umum. Studi pustaka dilakukan untuk melihat domain masalah ini secara lebih komprehensif dan ideal, karena pada aplikasi yang sedang berjalan mungkin telah dilakukan simplifikasi dan menghilangkan *requirement* yang penting. Secara bersamaan, domain permasalahan yang akan dibahas juga dipelajari sebagai bahan untuk mengetahui *requirement* agar bisa diimplementasikan ke dalam *pattern* yang akan disusun. Adapun prosedur yang akan dilakukan terdiri dari:

- a. Pengumpulan studi kasus yang nantinya dalam penelitian ini akan mengambil 3 (tiga) studi kasus. Pada tahapan ini pengumpulan data didapat dengan cara observasi, wawancara atau dalam bentuk kontribusi tulisan atau data yang sudah ada seperti dalam bentuk buku pedoman akademik. Dan informasi pada setiap studi kasus yang diperoleh akan digunakan sebagai referensi untuk pembuatan hasil observasi dalam bentuk narasi. Yang didokumentasikan dalam bentuk dokumen SRS.
- b. Untuk identifikasi permasalahan, dari seluruh studi kasus di cari kesamaan yang identik dari hasil observasi dengan cara melakukan pemetaan dalam bentuk tabulasi. Dari tabulasi ini akan didapat beberapa bagian yang sifatnya umum dengan menggunakan beberapa kriteria berdasarkan *basic flow* dan *Use Case* pada Dokumen SRS.
- c. Pemetaan *SRS Pattern* dilakukan setelah memperoleh bagian dari tabulasi di atas yang sifatnya *common* dan *intersection* dari tiga kasus yang sudah dianalisa, sehingga *SRS Pattern* dapat dibentuk.
- d. Tahapan akhir adalah dengan menganalisa hasil set *use case* yang diperoleh dari analisa sebelumnya dan merepresentasikannya ke dalam *template* pada *SRS Pattern*.

3. Requirement Pattern dan Software Requirement Specification (SRS) Pattern

3.1. Requirement Pattern

Banyak definisi yang digunakan untuk mendeskripsikan apa *Pattern* sebenarnya. Secara umum *Pattern* merupakan solusi dari konteks sebuah masalah. Salah satu definisi dari *Pattern* adalah definisi yang diberikan oleh Christopher Alexander, seorang arsitek yang pertama kali melontarkan penggunaan istilah "*Pattern*" ini: "*Each Pattern describes a problem which occurs over and over again in our environment, and then describes the core of the solution to that problem, in such a way that you can use the*

solution a million times over, without ever doing it the same way twice.” [8]. Sedangkan menurut Patricia Ferdinandi [9], *requirement pattern* merupakan kerangka untuk kebutuhan yang digunakan untuk mendukung kebutuhan produk dan meminimalisir gap yang bisa menyebabkan kegagalan proyek namun tetap mendukung *multiple* desain dan implementasi.

3.2. Analysis Pattern Untuk Aplikasi Course registration

Analysis pattern for course management [3], [4], yang diteliti oleh Fernandez merupakan salah satu penelitian yang dilakukan Fernandez dalam melakukan tahapan pembentukan sebuah *pattern* hasil dari *representasi semantic analysis pattern* sebelumnya yang diterapkan ke dalam sebuah kasus, yaitu *course management*. Pada penelitiannya Fernandez membagi dua komponen *Pattern*, yang salah satunya akan dijadikan acuan dalam penelitian ini. Dua komponen tersebut adalah *course registration* dan *grade management*. Yang mana nantinya kedua komponen ini akan direlasikan menjadi sebuah komponen utama, yaitu *course management*.

3.3. SRS Pattern Budiardjo [7]

Dalam penelitiannya Budiardjo menjelaskan bahwa penyusunan beberapa SRS untuk domain aplikasi yang sama, dengan dimulai dari awal, memperlihatkan adanya kesamaan (*commonality*) dan keaneka ragaman (*variability*). Solusi terhadap kesamaan tersebut tidak perlu dilakukan bila solusi tersebut sudah dikembangkan menjadi *pattern*. Dari hasil penelitiannya yang mengacu pada konsep Alexandrian, maka *representasi software requirement specification pattern* yang dibangun menghasilkan representasi 10 atribut yang terdapat pada Tabel 1 berikut.

Tabel 1. Pola Representasi SRS Pattern Budiardjo [7]

No	Atribut	Representasi
1	<i>Name</i>	Naratif
2	<i>Intent</i>	Naratif
3	<i>Motivation</i>	UML Activity Diagram
4	<i>Forces</i>	Naratif
5	<i>Context</i>	Naratif
6	<i>Solution</i>	Naratif
	<i>a. Structure</i>	UML Use-Case Diagram
	<i>b. Scenario</i>	Use Case Specification
7.	<i>Consequences</i>	Naratif
8.	<i>Known Uses</i>	Naratif
9.	<i>Related Requirement Pattern</i>	Naratif
10	<i>Also Known As</i>	Naratif

Berbeda dengan beberapa *pattern* yang ada, *template* ini menggunakan *use case diagram*, *activity diagram*, dan *use case specification* dalam merepresentasikan beberapa atributnya dan menyertakan *alternate flow* dalam representasinya yang pada Fernandez justru ditinggalkan. Sehingga *template pattern* ini sudah cukup memenuhi pada level requirement. Dan pada penelitian ini akan menggunakan representasi SRS *pattern* yang dikembangkan oleh Eko K. Budiardjo dengan dilakukan analisa studi kasus pada beberapa aplikasi SIAK.

4. Hasil dan Pembahasan

4.1. Analisa Use case Intersection Proses Course Registration

Dalam melakukan analisa *commonality* dan *intersection* dari proses *Course Registration* dilihat dari *use case* yang dari segi prosesnya hampir sama, dari ketiga kasus proses Registrasi Akademik pada aplikasi SIAK dapat dijelaskan pada Tabel 2 di bawah ini.

Tabel 2. Set Use Case Sitem Informasi Akademik

No	UC STT – PLN	UC FK UNRI	UC SIAK NG UI
1.	Pengisian KRS	Pengisian KRS	Pengisian IRS
2.	Perbaikan KRS	Proses Pembatalan KRS	Edit IRS
3.	Perbaikan KRS (batal)	Proses Penambahan KRS	Drop IRS
4.	Perbaikan KRS (tambah)		Add IRS
5.	Proses KSM		Approval IRS bermasalah
6.	Manajemen Jadwal Perkuliahan		Approval IRS Mahasiswa
7.			Melihat IRS
8.			Alokasi jadwal perkuliahan

Dari tabel di atas bisa diambil unsur yang saling beririsan dan irisan use case tersebut adalah:

- Proses Pengisian KRS/IRS
- Proses Perbaikan KRS/IRS
- Proses Penambahan KRS/IRS
- Proses Pembatalan KRS/IRS.

4.2. Set Use Case Commonality Dari Hasil Intersection Proses Registrasi Akademik

Dari tahapan-tahapan di atas, maka akan didapatkan set use case baru yang bisa dikatakan use case ini merupakan use case yang paling umum sebagai dasar kebutuhan dalam proses registrasi akademik, antara lain Proses Pengisian KRS/IRS, Proses Perbaikan KRS/IRS, Proses Penambahan KRS/IRS, dan Proses Pembatalan KRS/IRS.

Berdasarkan hasil tahapan analisis di atas, maka alternatif pattern yang bisa diidentifikasi diperlihatkan pada Tabel 3. Pada tabel tersebut disajikan aplikasi-aplikasi dan fungsi dalam Sistem Informasi Akademik yang bias diidentifikasi sebagai kandidat pattern yang dikembangkan. Beberapa aspek dalam memilih pattern yang akan dikembangkan di atas adalah factor keterkaitan antar komponen yang saling berhubungan berdasarkan analisa pada tahapan sebelumnya. Beberapa aspek di atas tidak secara keseluruhan yang merupakan proses utama dari proses registrasi akademik, namun secara proses menjadi prasyarat untuk masuk ke dalam proses registrasi akademik.

Tabel 3. Fungsi *course registration* pada SIAK dan kandidat *pattern* yang bisa diidentifikasi

Fungsi	Kemungkinan Pattern
Mendapatkan informasi data KRS/ IRS mahasiswa	Terkait dengan proses Manajemen Registrasi KRS/IRS
Menerima input data KRS/IRS mahasiswa	
Mengambil Informasi data KRS/IRS	
Mengambil informasi data KRS/IRS yang sudah terisi sebelumnya untuk dilakukan perubahan berupa edit dan add/drop pada pilihan matakuliah	Terkait dengan proses Perubahan KRS/IRS
Menangani proses approval KRS/IRS	Terkait dengan kegiatan Persetujuan KRS/IRS sebagai salah satu proses precondition
Menangani proses approval perubahan KRS/IRS	

4.3. Representasi Software Requirement Specification (SRS) Pattern Registrasi Akademik

Dari tahapan sebelumnya, diidentifikasi 3 buah pattern yang dihasilkan yang akan direpresentasikan dalam tabel 4, tabel 5 dan tabel 6 di bawah ini, yaitu:

a. Registrasi KRS/IRS

Registrasi KRS/IRS merupakan SRS use case yang menjelaskan proses pendaftaran KRS/IRS mahasiswa dan interaksi untuk mendapatkan informasi KRS/IRS mahasiswa. Registrasi KRS/IRS berkaitan dengan staff BAA/staff jurusan dalam melakukan pengelolaan data mahasiswa dan KRS/IRS mahasiswa. Dan juga berkaitan dengan proses mahasiswa dalam melakukan proses pengisian KRS/IRS dalam memilih matakuliah yang akan diikuti pada semester yang akan berjalan dan juga berkaitan dengan proses mahasiswa dalam mendapatkan data yang sudah terdaftar untuk melihat hasil status persetujuan dari sistem dan juga dari dosen pembimbing akademik.

Tabel 4. Identifikasi use case untuk Registrasi KRS/IRS

Use case Name	Use case Description	Actor
Manage Data Mahasiswa	UC ini menggambarkan proses untuk mengelola data mahasiswa yang sudah terdaftar sebagai mahasiswa	Staff BAA Staff Jurusan
Manage Data KRS/IRS Mahasiswa	UC ini menggambarkan proses untuk mengelola data KRS/IRS mahasiswa datanya sudah diisi sebelumnya	Staff BAA Staff Jurusan
Registrasi KRS/IRS	UC ini menjelaskan interaksi yang menggambarkan proses pendaftaran KRS dengan melakukan pengisian form KRS/IRS	Mahasiswa
Mengambil Informasi KRS/IRS Mahasiswa	UC ini menjelaskan proses actor melihat data KRS/IRS yang sudah diinput	Staff BAA Staff Jurusan Mahasiswa
Membuat Laporan KRS/IRS Mahasiswa	UC ini menjelaskan proses pembuatan laporan KRS/IRS mahasiswa sebagai bukti sah untuk mengikuti perkuliahan	Staff BAA Staff Jurusan

b. Perbaikan KRS/IRS

Perubahan KRS/IRS adalah proses yang muncul akibat adanya proses registrasi KRS/IRS. Pada proses ini menjelaskan proses perubahan KRS/IRS yang muncul akibat adanya data KRS/IRS yang bermasalah. Use case Perubahan data KRS/IRS mahasiswa akan muncul pada saat pemilihan mata kuliah yang dipilih ternyata bermasalah pada jadwal yang bentrok atau penuh karena kapasitas kelas yang sudah maksimal. Proses ini dapat dilakukan tanpa harus melakukan proses konsultasi dengan dosen pembimbing akademik. Sedangkan untuk dua proses lainnya, penambahan data KRS/IRS dan pengurangan data KRS/IRS mahasiswa diharuskan melakukan proses konsultasi akademik karena adanya permasalahan pada batasan- batasan ketentuan dalam pemilihan matakuliah tertentu.

Tabel 5. Identifikasi use case untuk Perbaikan KRS/IRS

Use case Name	Use case Description	Actor
Perubahan data KRS/IRS Mahasiswa	UC ini menjelaskan proses interaksi actor dalam melakukan perubahan data KRS/IRS tanpa adanya proses pengurangan dan penambahan matakuliah yang diambil	Mahasiswa
Penambahan data KRS/IRS Mahasiswa	UC ini menjelaskan proses interaksi actor dalam melakukan perubahan data KRS/IRS dalam melakukan proses penambahan matakuliah yang diambil	Mahasiswa
Pengurangan data KRS/IRS Mahasiswa	UC ini menjelaskan proses interaksi actor dalam melakukan perubahan data KRS/IRS dalam melakukan proses pengurangan matakuliah yang diambil	Mahasiswa

c. Persetujuan KRS/IRS

Pada proses persetujuan KRS/IRS mendeskripsikan proses dosen dalam melakukan verifikasi pengisian KRS/IRS mahasiswa dalam memilih matakuliah pada semester yang akan berjalan. Dosen akan melakukan pengecekan terhadap pilihan matakuliah yang diambil dengan melihat beberapa atasan seperti KHS (Kartu Hasil Studi) sebagai historical data nilai dengan mengambil acuan IPK dan IPS sebagai status perkembangan akademik mahasiswa untuk menentukan beban SKS yang bisa diambil, matakuliah prasyarat yang harus sudah diikuti pada matakuliah tertentu.

Tabel 6. Identifikasi use case untuk Persetujuan KRS/IRS

Use case Name	Use case Description	Actor
Approval registrasi KRS/IRS	UC ini menjelaskan proses interaksi menyetujui IRS yang telah diisi oleh Mahasiswa. Dosen yang dapat menyetujui hanya dosen yang selaku sebagai pembimbing akademis dan hanya bisa menyetujui mahasiswa bimbingan akademisnya.	Dosen
Approval perubahan KRS/IRS	UC ini menjelaskan proses interaksi menyetujui IRS yang telah diisi oleh Mahasiswa. Dosen yang dapat menyetujui hanya dosen yang selaku sebagai pembimbing akademis dan hanya bisa menyetujui mahasiswa bimbingan akademisnya.	Dosen

5. Kesimpulan

Software Requirement Specification (SRS) Pattern merupakan *Pattern* pada level *Requirement Pattern*, sehingga representasi yang diberikan hanya merepresentasikan pada tahapan *requirement* dan analisis yang lebih dekat pada aspek model fungsi aplikasi untuk menghasilkan spesifikasi kebutuhan perangkat lunak. *SRS Pattern* untuk suatu aplikasi bisa diterapkan untuk membuat model inisial dengan cepat, khususnya diimplementasikan dalam ruang lingkup yang sama. Representasi template *SRS Pattern* dari Budiardjo yang mengacu pada *Analysis Pattern* Fernandez bisa digunakan untuk merepresentasikan *requirement* secara efektif pada domain aplikasi Sistem Informasi Akademik dengan melakukan perbandingan beberapa aspek keumuman (*commonality*) pada beberapa aplikasi yang sama. *SRS Pattern* bisa direpresentasikan secara efektif dengan notasi UML yang bisa mendeskripsikan model konseptual dan objek-objek aplikasi Sistem Informasi Akademik.

Referensi

- [1] Fernandez, Eduardo B, *Good Analysis As the Basis For Good Design And Implementation Of Object Oriented Systems*, Dept. of Computer Science and Eng. , Florida Atlantic University Boca Raton, FL, 1997.
- [2] Fernandez, Eduardo B, *Building Systems Using analysis Pattern Processes*. The 3rd Int. Soft. Architecture Workshop (ISAW3), Orlando, FL, November 1998. 37-40.
- [3] Fernandez, Eduardo B, and X. Yuan, *An Analysis Pattern for Reservation and Use of Reusable Entities*, Dept. of Computer Science and Eng. , Florida Atlantic University Boca Raton, FL, 1999.
- [4] Fernandez, Eduardo B, and X. Yuan, *An Analysis Pattern for Course Management*, Dept. of Computer Science and Eng. , Florida Atlantic University Boca Raton, FL, 2001.
- [5] Andriyani, Y., Budiardjo, Eko K., *Penyusunan Software Requirement Specification Use Case (SRS UC) Sistem Informasi Akademik STT PLN Dengan Pendekatan Reverse Engineering*, Technical Report Fakultas Ilmu Komputer Universitas Indonesia, UI – CS – 2008 – xx , Juni 2008.
- [6] Andriyani, Y., Budiardjo, Eko K., *Penyusunan Software Requirement Specification Use Case (SRS UC) Sistem Informasi Akademik Fakultas Kedokteran Universitas Riau (FK UNRI) Dengan Pendekatan Reverse Engineering*, Technical Report Fakultas Ilmu Komputer Universitas Indonesia, UI – CS – 2008 – xx , Juni 2008.
- [7] Budiardjo, Eko K, Pola Representasi Software Requirement Specification (SRS) Pattern, *Jurnal Sistem Informasi MTI -UI*, 2008, 4(1).
- [8] Alexander, C. A Pattern Language: Towns, Buildings, Construction (Cess Center for Environmental. New York: Oxford University Press, 1977.
- [9] Ferdinandi, Patricia. A Requirements Pattern-Succeeding in the Internet Economy. New York: Addison-Wesley, 2002.